10/659066 Aeg 6/4/10

Please amend the paragraph beginning on page 72, line 1, as follows:

To examine the aliasing effect described above, and to determine what an appropriate number of buckets ~~110~~ 106 should be, FIG. 30 shows the sum of the differences in the bucket weights found between all sequential intervals of execution. The y-axis shows the sum total of differences for each program. This is calculated by summing the differences between the buckets ~~110~~ 106 captured for interval i and i-1 for each interval i in the program. The x-axis is the number of distinct buckets used. All of the results are compared to the ideal case of using an infinite number of buckets ~~110~~ (or one for each separate basic block) to create the signature. On the program gcc, for example, the total sum of differences with 32 buckets was 72% of that captured with an infinite number of buckets. In general, 32 buckets has been found to be enough to distinguish between two phases.

Please amend the paragraph beginning on page 83, line ~~1~~ 3, as follows: Aeg 6/4/10

Just-in-time (JIT) systems are assisted, as efficient JIT systems can be built to guide when to spend time on optimizing code. By using a hardware-independent metric for the component such as the code executed, analysis may be performed in a very short amount of time, on the order of how long it takes to execute the program itself, using a very fast high level code profiler. ~~Reoptimiztion~~ Reoptimization of a program can be expedited by determining when to perform the reoptimization.